



02

Actualizando datos #1

Transcripción

[00:00] Bien, vamos a continuar, ya sabemos que nuestro put digamos, ya está cableado como decimos, el request del cliente ya está llegando a mi servidor, entonces ahora vamos a proceder a implementarlo. ¿Cómo?

[00:14] Primero el DTO va a cambiar porque repitiendo lo que vimos en el video anterior, aquí los parámetros son obligatorios, pero por requerimiento yo no debo permitir actualizar email, especialidad y documento. Email, especialidad y documento son obligatorios aquí. Yo no debería enviarlos. Entonces, para no hacer tanto problema, yo voy a poner aquí un `datosActualizarMedico`.

[00:40] Este va a ser mi nuevo DTO. Vamos a poner el mismo nombre. Y voy a crear otro Java record. Para crear mi DTO. Venimos aquí, cree el record dentro del paquete médico. Perfecto y le damos okay. Aun no lo agregamos a Git. ¿Y qué es lo que vamos a aceptar aquí? Bueno, ¿qué podemos actualizar? Vamos al requerimiento y solo es nombre, documento y dirección.

[01:10] Venimos aquí y le damos `String nombre`, `String documento` y `Dirección`. Y para dirección, como las reglas de negocio no han cambiado para dirección, porque para la dirección solo sabemos que se puede actualizar, pero obviamente una dirección hay ciertos datos que debe tener como calle, distrito, ciudad número o complemento.

[01:43] Entonces dirección no va a cambiar en reglas del negocio, todos lo vamos a mantener tal cual, con las mismas validaciones. Entonces tenemos

dirección, documento, nombre y como ya lo vimos en la clase anterior, para actualizar necesitamos saber el id y le vamos a agregar un Long id.

[01:59] Ahora el único parámetro que debería ser obligatorio debería ser el id, porque sin el id yo simplemente no sé qué elemento es el que yo quiero actualizar. Entonces yo le voy a poner un NotNull. ¿Por qué? Porque es un long. El NotBlank sería para los que son strings, pero como es un long yo le pongo NotNull.

[02:22] Con eso es más seguro que el id nunca va a venir vacío y si es que viene vacío, entonces retornable un bad request porque yo no sé qué datos quieres que actualice. Ahora venimos al MedicoController, vemos que el método ya compila ¿y cuál es el primer paso que debemos hacer aquí?

[02:42] Por ejemplo, ustedes saben que quieren actualizar a Diego López con el id 5. Pero yo lo tengo aquí en el listado. ¿Qué es lo que yo tendría que hacer aquí para obtenerlo de la base de datos? Bien primero, entonces lo que voy a crear aquí es una instancia de médico.

[03:05] Y aquí yo voy a llamar a mi repositorio, MedicoRepository.getReferenceById. ¿Por qué? Porque como en eso yo voy a decirle datosActualizarMedico.Id, porque lo está recibiendo por el cliente, el cliente me está enviando el id, entonces yo digo: “déjame buscar ese médico en mi base de datos para obtenerlo y yo voy a obtener esa entidad médico”.

[03:31] Nuevamente, al igual que el final, el getReferenceById, si vamos al repositorio, no hay nada en mi repositorio. Esto es porque simplemente ese método existe en la interfaz JpaRepository, de la cual nosotros estamos extendiendo. Es una más de las ventajas de Spring, ya podemos tener esta query y no vamos a hacer nada nosotros.

[03:51] ¿Qué más vamos a hacer? Bueno, vamos a decirle medico.actualizarDatos vamos a crear este método dentro de médico y actualizar datos debe recibir datosActualizarMedico. Entonces vamos a crearlo

automáticamente, Spring lo puede hacer por nosotros. ¿Y qué es lo que vamos a hacer aquí?

[04:14] Aquí vamos a decirle entonces que el nombre del médico me lo iguale a lo que está llegando del DTO, bien parecido a lo que está aquí. Entonces yo puedo copiar, por ejemplo, entre los datos que yo puedo actualizar están nombre, documento y dirección. Entonces voy a copiar esto.

[04:32] Le voy a decir nombre, copia aquí, lo mismo para documento. Voy por ahora copiando en DTO, nombre, documento, aquí también voy a obtener el documento y para la dirección yo voy a hacer exactamente un método para actualizar los datos de dirección, como simplemente `this.direccion = direccion.actualizarDatos`.

[05:10] ¿Y como parámetro qué creen que va ahí? Va a ir `datosActualizarMedico.direccion`. Como este método no existe, de igual forma yo lo voy a crear en dirección y simplemente vamos a copiar esto. Y ustedes pueden pensar entonces ya está, vamos a hacer un `return direccion`. Ya está. Vamos a retomar el mismo objeto porque es lo que estamos actualizando.

[05:43] Y aquí también, pues ese es un void. Entonces todo bien, no necesitamos nada más. Y al inicio ustedes me pueden decir: “Okay, Diego, suficiente con eso”. Pero aquí viene una consideración muy especial porque recuerden, yo puedo actualizar o uno o todos los parámetros, pero por ejemplo, ¿qué pasa si yo solo quiero modificar el documento?

[06:06] Entonces, en mi payload que yo voy a enviar aquí, por ejemplo, yo puedo decirle, puedo copiar esto y en el request put solamente mandarle un nombre. Si yo hago eso, entonces documento, llega null, dirección también llega null, y de la forma cómo está mapeado ahora lo que él va a hacer es reemplazarlo por null.

[06:31] ¿Entonces, qué puedo hacer? Preguntar si el parámetro está llegando vacío o no, entonces lo voy a decir `if`, abro los paréntesis. ¿Y aquí la condición

cuál va a ser? If (`datosActualizarMedico.nombre()` es diferente de null. ¿Entonces, qué es lo que yo voy a hacer? En ese momento recién voy a actualizar los datos.

[07:00] De igual manera, para cada uno de estos parámetros. Yo ya lo borro de aquí. Y nuevamente voy a preguntarle acá si nuevamente abro y voy a hacer exactamente la misma validación. Si documento es diferente de null y no lleva nulo, es porque yo estoy mandando una información en ese documento, entonces ahí actualízame el documento.

[07:32] No voy a comparar si es que está vacío también porque ya se sobreentiende que yo no voy a poner a propósito este parámetro en el front end vacío. Finalmente, el último if que vamos a hacer aquí. Venimos acá, abrimos y vamos a preguntarle, pues si dirección, `datosActualizarMedico.dirección` es diferente de null.

[08:00] Y con esto, simplemente yo corto, pego aquí y ya tengo mi método. Con esto ya está mi método para actualizar los datos dependiendo, pues lo que está llegando aquí. Para lo que es actualizar datos de dirección, yo estoy igualando todo porque mis parámetros de dirección vemos que todos están siendo aquí considerados.

[08:25] No voy a ir a mi DTO de dirección, que sería en el `datosRegistroMedico`, voy a `DatosDirección`. Y automáticamente él ya me valida que ninguno de estos datos llegue vacío. Entonces, por ese lado yo estoy tranquilo, por ese lado no me voy a preocupar de aplicar la misma validación porque ya a nivel de DTO ya lo estoy haciendo.

[08:50] Entonces yo ya sé que cuando actualice los datos de la dirección va a venir una dirección válida y yo puedo realizar, puedo igualar estos parámetros automáticamente. Entonces sin más que decir, vamos a probar el método.

[09:05] Guardamos aquí y ahora llega la hora de modificar nuestro request, entonces voy a venir aquí a mi new request y voy a crear el payload que yo

quiero enviar. Selecciono JSON, perfecto y yo voy a agarrar un método del listado, ya dije que yo quiero actualizar a Diego López con el id 5, entonces voy aquí, de pasada voy a cambiarle el nombre, le voy a poner a Actualizar Medico, solo para que quede más entendible.

[09:36] Renombramos y pegamos el Jason, pero como les dije, yo quiero solamente especialidad no se puede actualizar, entonces se las voy a borrar. Payload. Y solamente quiero digamos actualizarle el nombre, no el documento ni el email. El email no se puede tampoco.

[09:56] Entonces yo le voy a decir, quiero actualizar el nombre de Diego López porque como tengo dos Diego López a él lo vamos a llamar por su segundo nombre, que puede ser Luis por ejemplo. Él se llama Diego Luis López entonces ahora para que se diferencien entre ellos, va a ser Luis.

[10:13] Ahora, yo entro aquí a mi controller, voy a MedicoController ya está el actualizar médico, ya está el datosActualizarMedico como DTO, vamos a ver si mi método funciona, le doy enviar y me salió un 500 internal server error. Y me dice InvalidDataAccessException, el id no puede ser null.

[10:35] ¿Qué quiere decir? Que el id que tiene aquí no ha sido detectado por Spring a pesar de que yo lo estoy enviando aquí como no nulo. ¿Por qué? ¿Por qué es este problema?